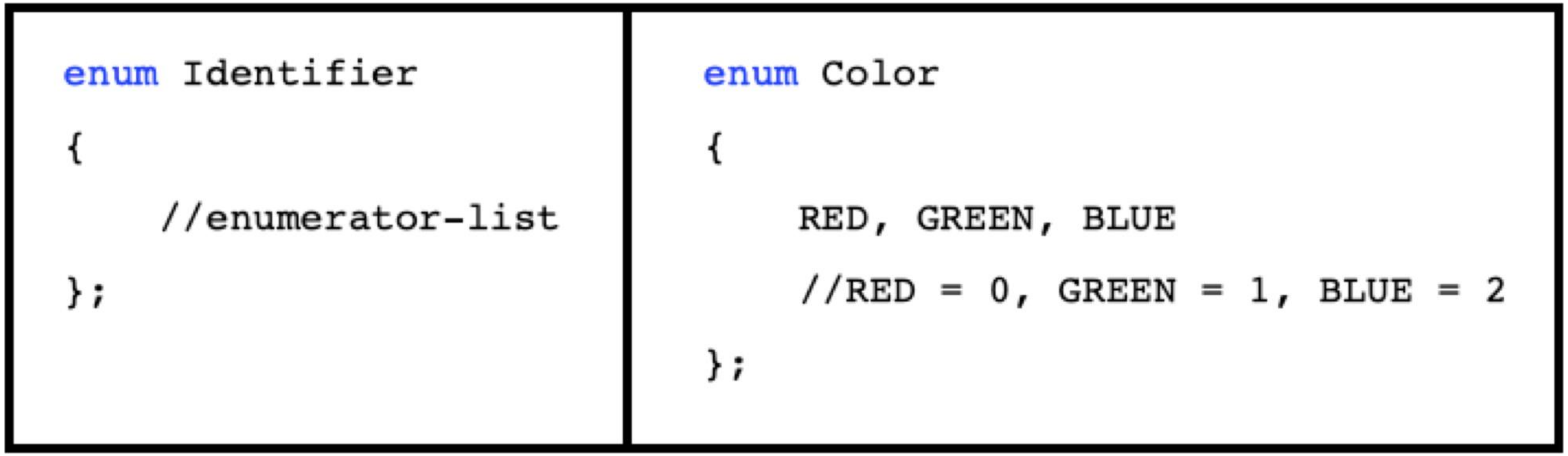
Core Java

Akshita Chanchlani

Enum In C/C++ Programming language.

- According ANSI C standard, if we want to assign name to the integer constant then we should use enum.
- Enum helps developer to improve readability of source code.
- enum is keyword in C. Let us consider syntax of enum:

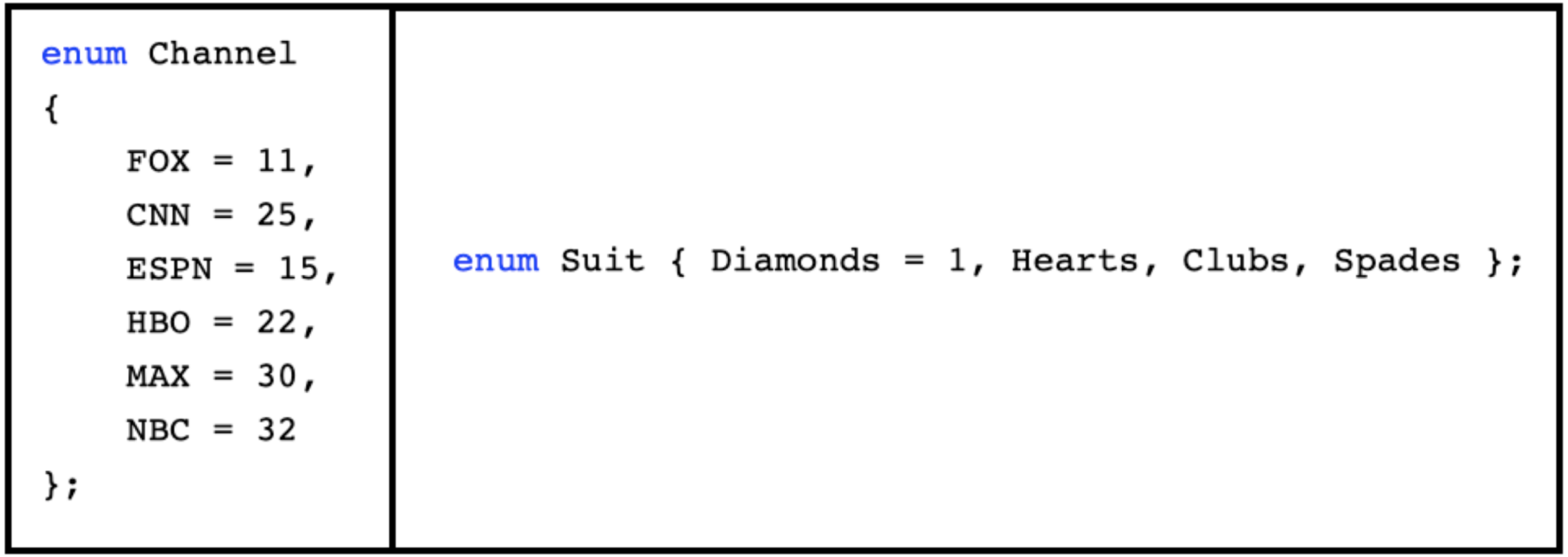


```
enum Identifier
{
    //enumerator-list
};
```

```
enum Color
{
    RED, GREEN, BLUE
    //RED = 0, GREEN = 1, BLUE = 2
};
```

Enum In C/C++ Programming language.

- By default, the first enumeration-constant is associated with the value 0. The next enumeration-constant in the list is associated with the value of (constant-expression + 1), unless you explicitly associate it with another value.



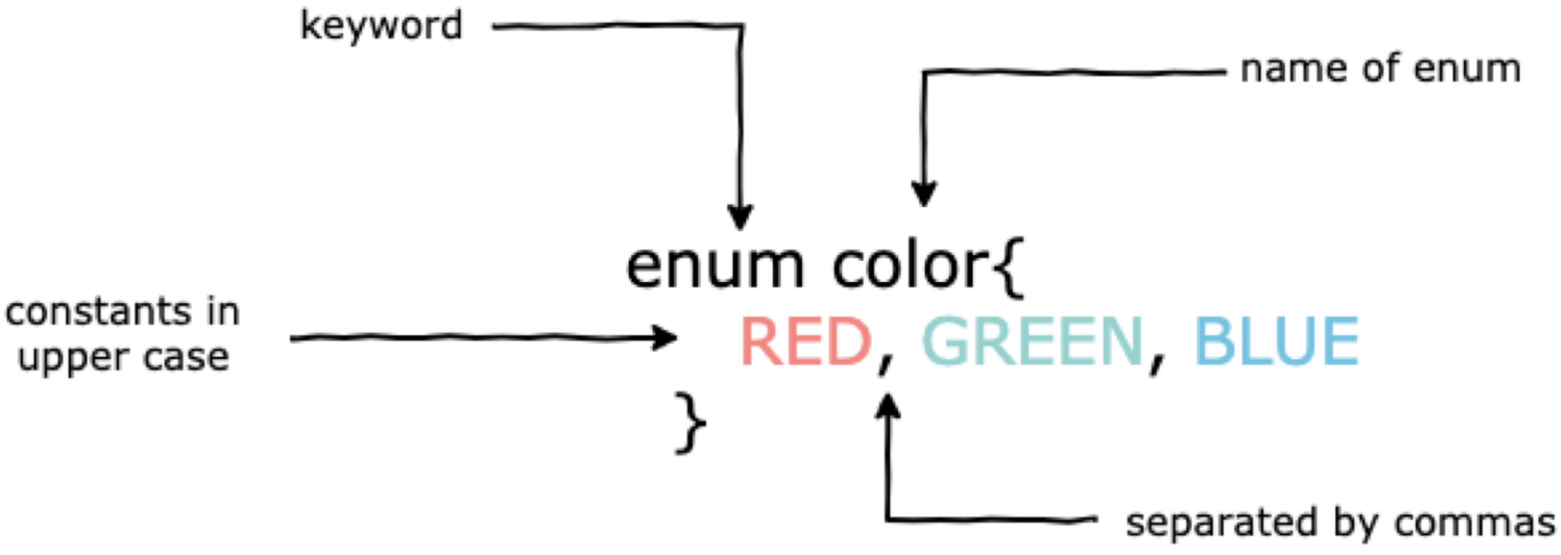
```
enum Channel
{
    FOX = 11,
    CNN = 25,
    ESPN = 15,
    HBO = 22,
    MAX = 30,
    NBC = 32
};
```

```
enum Suit { Diamonds = 1, Hearts, Clubs, Spades };
```

- constant-expression must have int type and can be negative.

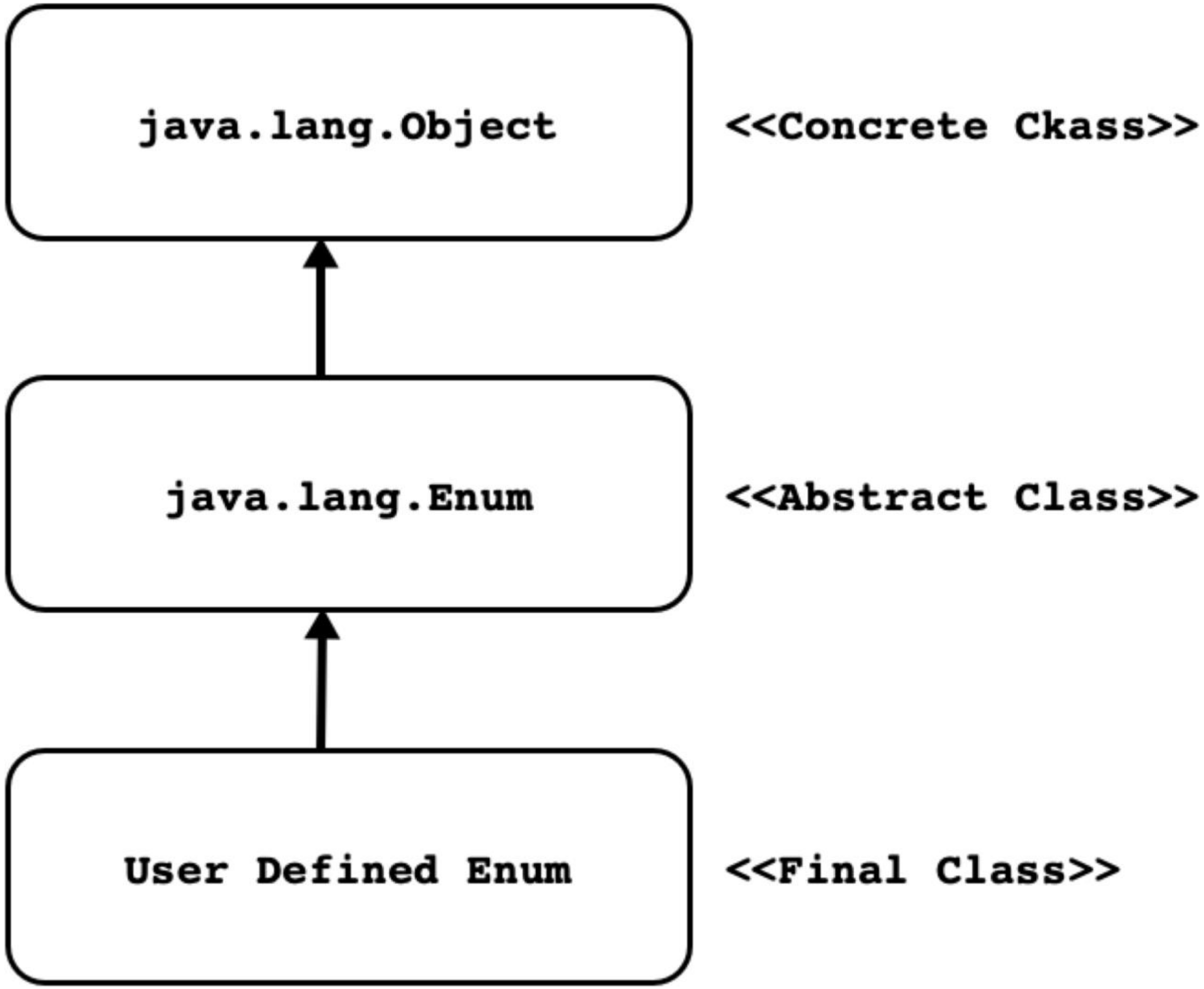
Enum In Java Programming language.

- An enum is a class that represents a group of constants.
- **Enum keyword** is used to create an enum. The constants declared inside are separated by a comma and should be in upper case.



```
enum color{
  RED, GREEN, BLUE
}
```

Enum Class Hierarchy



java.lang.Object <<Concrete Ckass>>

java.lang.Enum <<Abstract Class>>

User Defined Enum <<Final Class>>

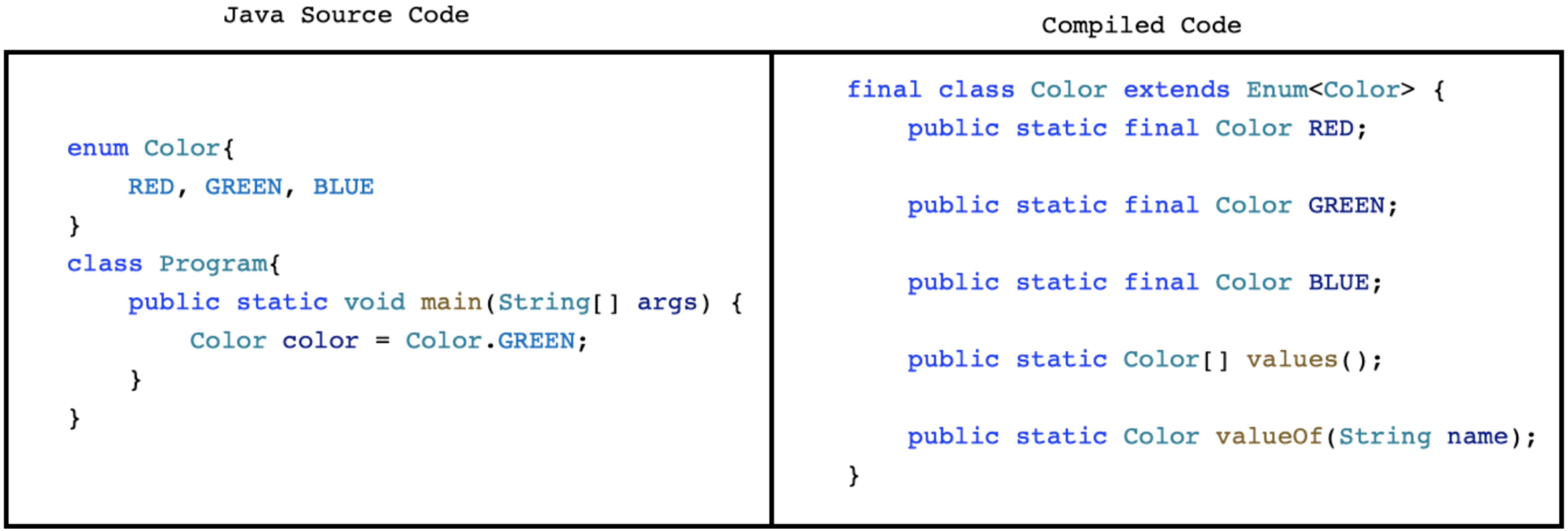
Enum API

- **Following are the methods declared in java.lang.Enum class:**

String	name()Returns the name of this enum constant, exactly as declared in its enum declaration.
int	ordinal()Returns the ordinal of this enumeration constant (its position in its enum declaration, where the initial constant is assigned an ordinal of zero).
String	toString()Returns the name of this enum constant, as contained in the declaration.
static <T extends Enum<T>> T	valueOf(Class<T> enumType, String name)Returns the enum constant of the specified enum type with the specified name.

Sole constructor : Programmers cannot invoke this constructor. It is for use by code emitted by the compiler in response to enum type declarations.

Enum for the compiler



Java Source Code

```
enum Color{
    RED, GREEN, BLUE
}
class Program{
    public static void main(String[] args) {
        Color color = Color.GREEN;
    }
}
```

Compiled Code

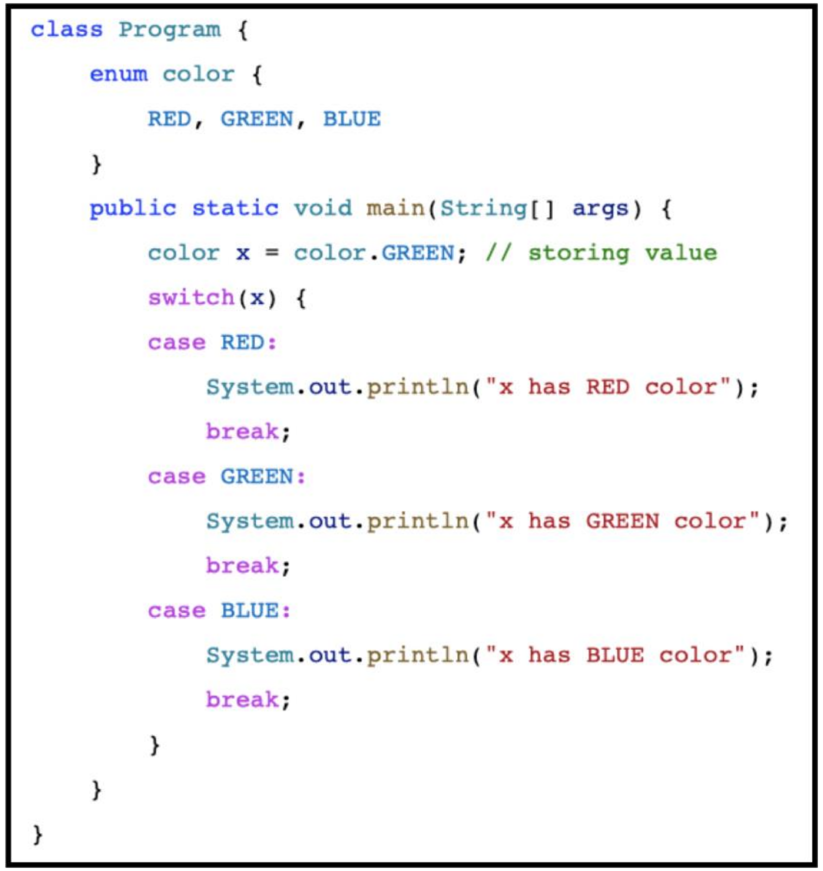
```
final class Color extends Enum<Color> {
    public static final Color RED;

    public static final Color GREEN;

    public static final Color BLUE;

    public static Color[] values();

    public static Color valueOf(String name);
}
```

Properties of enum

1. Similar to a class, an enum can have objects and methods. The only difference is that enum constants are public, static and final by default. Since it is final, we can’t extend enums
2. It cannot extend other classes since it already extends the java.lang.Enum class.
3. It can implement interfaces.
4. The enum objects cannot be created explicitly and hence the enum constructor cannot be invoked directly.
5. It can only contain concrete methods and no abstract methods.

Application of enum

1. enum is used for values that are not going to change e.g. names of days, colors in a rainbow, number of cards in a deck etc.
2. enum is commonly used in switch statements and below is an example of it:



```
class Program {
    enum color {
        RED, GREEN, BLUE
    }
    public static void main(String[] args) {
        color x = color.GREEN; // storing value
        switch(x) {
        case RED:
            System.out.println("x has RED color");
            break;
        case GREEN:
            System.out.println("x has GREEN color");
            break;
        case BLUE:
            System.out.println("x has BLUE color");
            break;
        }
    }
}
```

Thank You.